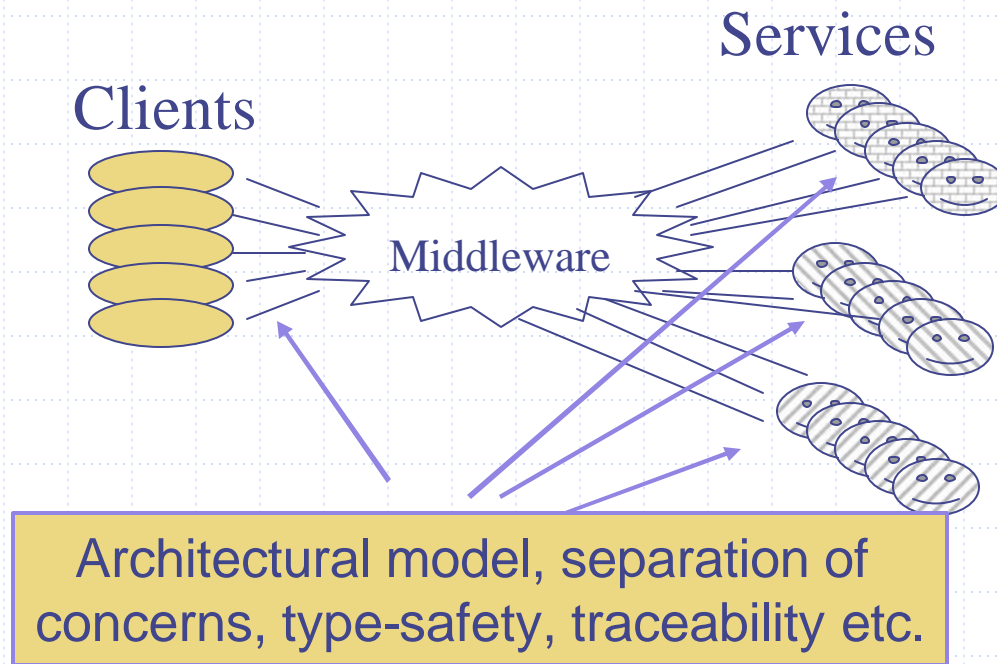


Evolving Distributed Heterogeneous Systems

Premkumar Devanbu &
Eric Wohlstadter,
Dept. of Computer Science,
University of California, Davis.

<http://castle.cs.ucdavis.edu>

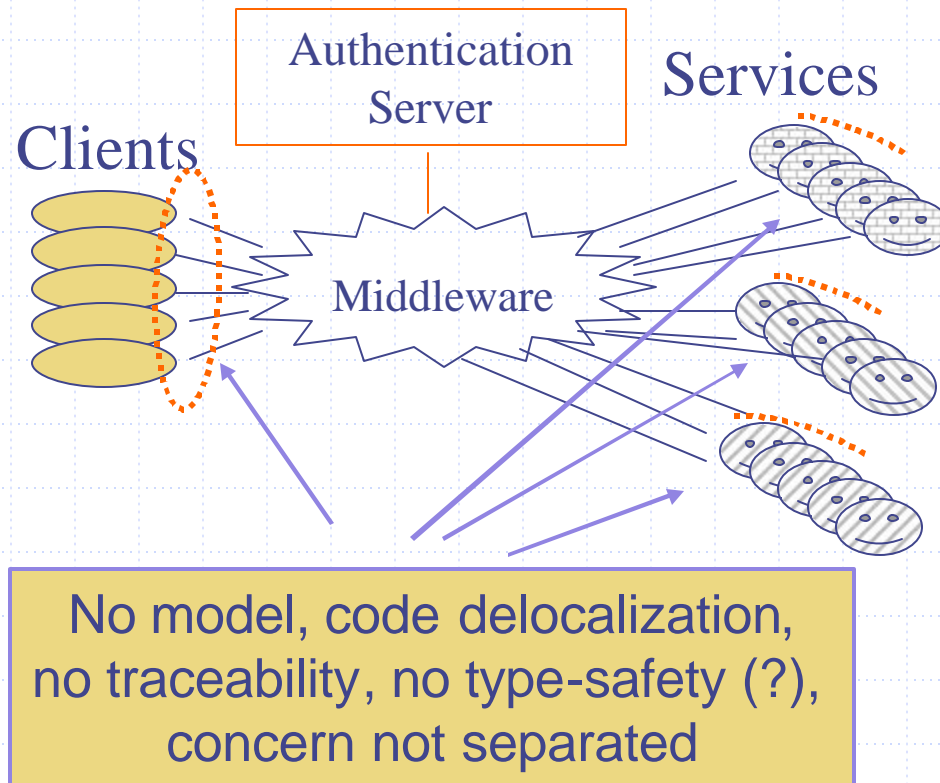
A distributed system.



For example:

- **Clients:** *patients:*
- **Services:** *doctors, pharmacists, and insurers.*

Now add Security



- ◆ *Non-local policy, e.g., pick one doctor only for “psychotropic” drugs.*
- ◆ *Fine-grained policies: e.g., state-full, w/delegation, data-dependent etc.*

Now add another feature

(Fault-tolerance, Logging...etc)

- ◆ Same problem as before, scattered implementation.
- ◆ *but worse...* features interact. Where is the interaction?

What we hope for.

- ◆ Separation of concerns:
 - A coherent unit where each feature lives
 - Ditto where interactions are handled.
- ◆ Architectural Modeling and Code generation
 - Architectural modeling of non-functional features
 - Architectural modeling of feature interactions
 - Traceability to implementation.
- ◆ Embrace Heterogeneity
 - Interoperability
 - Opportunistic use of available language tools.
- ◆ Type-safe & Conceptually well-founded
 - To enable static analysis, optimization, etc.

Some Potential Solutions.

- ◆ AOP.
- ◆ Wrappers.
- ◆ Monadic-style programming.
- ◆ Reflective techniques.
- ◆ Dynamic Techniques.

... all help, but not completely. We need synthesis, not only in better tools, but also for better methods.

If we solve this problem....

- ◆ Non-functional features easier to design, implement, and evolve.
- ◆ Feature-interactions ditto.
- ◆ Better-quality and better-performing systems.

What we hope for.

- ◆ Separation of concerns.
- ◆ Controlled, separable, composition of concerns.
- ◆ Typesafety
- ◆ Architectural modeling and traceability
- ◆ Inter-operability (legacy, binary-only etc).
- ◆ Opportunistic use of existing tools.

Difficulties

- ◆ Implementation de-localized (even scattered), but must be co-ordinated.
- ◆ Possibly different languages.
- ◆ Maintenance:
 - Tracking policy changes
 - Or even just fixing bugs.